Naive Bayes with Correlation Factor for Text Classification Problem

Juntao Duan

(Joint work with Jiangning Chen, Zhibo Dai, Heinrich Matzinger, Ionel Popescu)

Department of mathematics
Georgia Institute of Technology

jt.duan@gatech.edu

December 18, 2019

Overview

1. Motivation: text classification
2. Naive Bayes (NB) classifier
   - Basics of NB classifier
   - Error analysis
3. Correlation factor for NB
   - Insights from Neural net
   - Correlation factor
4. Simulation
   - Small training set
   - Large training set

Table of Contents

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Classification methods:

- Neural networks (well-tuned)

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Classification methods:

- Neural networks (well-tuned) $>$ Ensemble methods (Random forest, Boosting)

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Classification methods:

- Neural networks (well-tuned) $>$ Ensemble methods (Random forest, Boosting) $>$ SVM, KNN, Naive Bayes $>$ Logistic regression

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Classification methods:

- Neural networks (well-tuned) $>$ Ensemble methods (Random forest, Boosting) $>$ SVM, KNN, Naive Bayes $>$ Logistic regression
- Why neural networks are more effective?

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Classification methods:

- Neural networks (well-tuned) $>$ Ensemble methods (Random forest, Boosting) $>$ SVM, KNN, Naive Bayes $>$ Logistic regression
- Why neural networks are more effective?
- Can we migrate some features of neural nets to simpler models?

Motivation

Tasks:

- Emails $\rightarrow$ spam and non-spam
- Reviews of product (Amazon, IMDB, Yelp) $\rightarrow$ fake and genuine; positive, neutral and negative

Classification methods:

- Neural networks (well-tuned) $>$ Ensemble methods (Random forest, Boosting) $>$ SVM, KNN, Naive Bayes $>$ Logistic regression
- Why neural networks are more effective?
- Can we migrate some features of neural nets to simpler models?
    - Neural networks v.s. Naive Bayes

Table of Contents

Naive Bayes (NB) classifier for text classification

Assume

- Sample document set S, word dictionary $(word_1, \cdots, word_v)$.

Naive Bayes (NB) classifier for text classification

Assume

- Sample document set S, word dictionary $(word_1, \cdots, word_v)$.
- Any document $d \in S$, we denote frequecy of $word_1 \in d$ as x_1. Then d has words frequecy

$$\{x_1, x_2, \cdots, x_v\}$$

Naive Bayes (NB) classifier for text classification

Assume

- Sample document set S, word dictionary $(word_1, \cdots, word_v)$.
- Any document $d \in S$, we denote frequecy of $word_1 \in d$ as x_1. Then d has words frequecy

$$\{x_1, x_2, \cdots, x_v\}$$

- Class set C with k different classes:

$$C = \{C_1, C_2, ..., C_k\}.$$

Naive Bayes (NB) classifier for text classification

Assume

- Sample document set S, word dictionary $(word_1, \cdots, word_v)$.
- Any document $d \in S$, we denote frequecy of $word_1 \in d$ as x_1. Then d has words frequecy

$$\{x_1, x_2, \cdots, x_v\}$$

- Class set C with k different classes:

$$C = \{C_1, C_2, ..., C_k\}.$$

- For each document d, define its label vector

$$y(d) = (y_1(d), y_2(d), \cdots, y_k(d))$$

If document d is in class C_i, we have $y_i(d) = 1$, else are 0.

Basics of Naive Bayes

- Each class C_i $(1 \leq i \leq k)$ with words probability vector $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$. Each $P(word_j \in d | d \in C_i) = \theta_{i_j}$ and they satisfy: $\sum_{j=1}^{v} \theta_{i_j} = 1$.

Basics of Naive Bayes

- Each class C_i $(1 \leq i \leq k)$ with words probability vector $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$. Each $P(word_j \in d | d \in C_i) = \theta_{i_j}$ and they satisfy: $\sum_{j=1}^{v} \theta_{i_j} = 1$.
- Assuming independence of the words, the most likely class for a document d is computed as:

$$label(d) = \operatorname*{argmax}_i P(d \in C_i | d = (x_1, \cdots x_v))$$

Basics of Naive Bayes

- Each class C_i $(1 \leq i \leq k)$ with words probability vector $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$. Each $P(word_j \in d | d \in C_i) = \theta_{i_j}$ and they satisfy: $\sum_{j=1}^{v} \theta_{i_j} = 1$.
- Assuming independence of the words, the most likely class for a document d is computed as:

$$\begin{aligned} label(d) &= \operatorname*{argmax}_i P(d \in C_i | d = (x_1, \cdots x_v)) \\ &= \operatorname*{argmax}_i P(C_i) P(d | d \in C_i) \end{aligned}$$

Basics of Naive Bayes

- Each class C_i $(1 \leq i \leq k)$ with words probability vector $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$. Each $P(word_j \in d | d \in C_i) = \theta_{i_j}$ and they satisfy: $\sum_{j=1}^{v} \theta_{i_j} = 1$.
- Assuming independence of the words, the most likely class for a document d is computed as:

$$\begin{aligned} label(d) &= \underset{i}{\operatorname{argmax}} P(d \in C_i | d = (x_1, \cdots x_v)) \\ &= \underset{i}{\operatorname{argmax}} P(C_i) P(d | d \in C_i) \\ &= \underset{i}{\operatorname{argmax}} P(C_i) \prod_{j=1}^{v} (\theta_{i_j})^{x_j} \end{aligned}$$

Basics of Naive Bayes

- Each class C_i $(1 \leq i \leq k)$ with words probability vector $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$. Each $P(word_j \in d | d \in C_i) = \theta_{i_j}$ and they satisfy: $\sum_{j=1}^{v} \theta_{i_j} = 1$.
- Assuming independence of the words, the most likely class for a document d is computed as:

$$\begin{aligned}
label(d) &= \underset{i}{\operatorname{argmax}} P(d \in C_i | d = (x_1, \cdots x_v)) \\
&= \underset{i}{\operatorname{argmax}} P(C_i) P(d | d \in C_i) \\
&= \underset{i}{\operatorname{argmax}} P(C_i) \prod_{j=1}^{v} (\theta_{i_j})^{x_j} \\
&= \underset{i}{\operatorname{argmax}} \log P(C_i) + \sum_{j=1}^{v} x_j \log \theta_{i_j}.
\end{aligned}$$

Basics of Naive Bayes

- Each class C_i $(1 \leq i \leq k)$ with words probability vector $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$. Each $P(word_j \in d | d \in C_i) = \theta_{i_j}$ and they satisfy: $\sum_{j=1}^{v} \theta_{i_j} = 1$.
- Assuming independence of the words, the most likely class for a document d is computed as:

$$
\begin{aligned}
label(d) &= \operatorname*{argmax}_i P(d \in C_i | d = (x_1, \cdots x_v)) \\
&= \operatorname*{argmax}_i P(C_i) P(d | d \in C_i) \\
&= \operatorname*{argmax}_i P(C_i) \prod_{j=1}^{v} (\theta_{i_j})^{x_j} \\
&= \operatorname*{argmax}_i \log P(C_i) + \sum_{j=1}^{v} x_j \log \theta_{i_j}.
\end{aligned}
$$

How to find θ_{i_j}?

Estimating: maximum likelihood

- For a given class C_i we estimate $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$

$$\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j}.$$

Estimating: maximum likelihood

- For a given class C_i we estimate $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$

$$\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j}.$$

-

$$\begin{aligned} \max \quad & \log L(C_i, \theta) \\ \text{subject to :} \quad & \sum_{j=1}^{v} \theta_{i_j} = 1; \quad \theta_{i_j} \geq 0 \end{aligned}$$

Estimating: maximum likelihood

- For a given class C_i we estimate $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$

$$\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j}.$$

-

$$\begin{array}{rl} \max & \log L(C_i, \theta) \\ \text{subject to :} & \sum_{j=1}^{v} \theta_{i_j} = 1; \quad \theta_{i_j} \geq 0 \end{array}$$

- By Lagrange multiplier,

$$\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}. \tag{1}$$

Estimating: maximum likelihood

- For a given class C_i we estimate $\theta_i = (\theta_{i_1}, \theta_{i_2}, ..., \theta_{i_v})$

$$\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j}.$$

-

$$\begin{aligned} \max \qquad & \log L(C_i, \theta) \\ \text{subject to :} \qquad & \sum_{j=1}^{v} \theta_{i_j} = 1; \quad \theta_{i_j} \geq 0 \end{aligned}$$

- By Lagrange multiplier,

$$\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}. \tag{1}$$

How good is the estimator?

Error analysis

Theorem

Assume we have normalized length of each document, that is: $\sum_{j=1}^{V} x_j = m$ *for all documents* $d \in S$, *the estimator* (1) *satisfies following properties:*

1. $\hat{\theta}_{i_j}$ *is unbiased.*
2. $E[|\hat{\theta}_{i_j} - \theta_{i_j}|^2] = \frac{\theta_{i_j}(1-\theta_{i_j})}{|C_i|m}$.

Error analysis

Theorem

Assume we have normalized length of each document, that is: $\sum_{j=1}^{v} x_j = m$ *for all documents* $d \in S$*, the estimator* (1) *satisfies following properties:*

1. $\hat{\theta}_{i_j}$ *is unbiased.*
2. $E[|\hat{\theta}_{i_j} - \theta_{i_j}|^2] = \frac{\theta_{i_j}(1-\theta_{i_j})}{|C_i|m}$.

More words ($m \nearrow$), more documents ($|C_i| \nearrow$) $\Rightarrow$ less error.

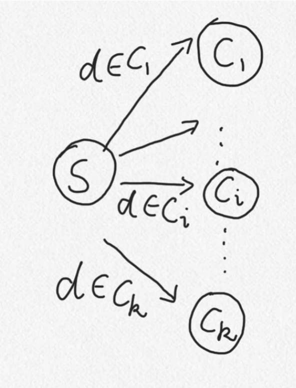
Table of Contents

Insights from Neural net

Naive bayes: $\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}$

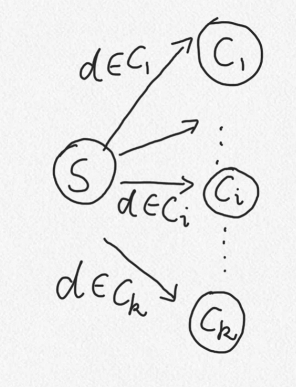
Insights from Neural net

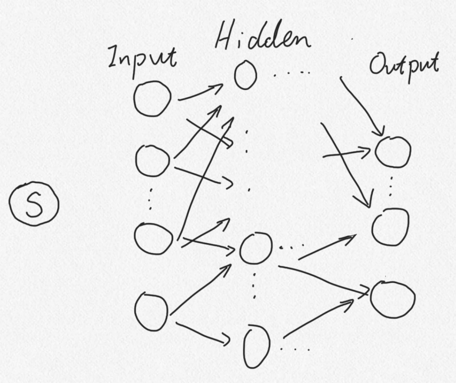
Naive bayes: $\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}$



Insights from Neural net

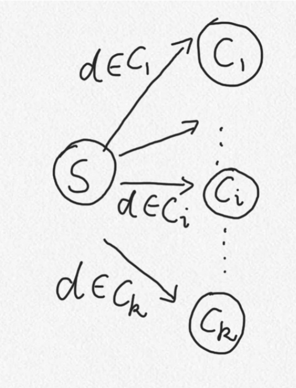
Naive bayes: $\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}$

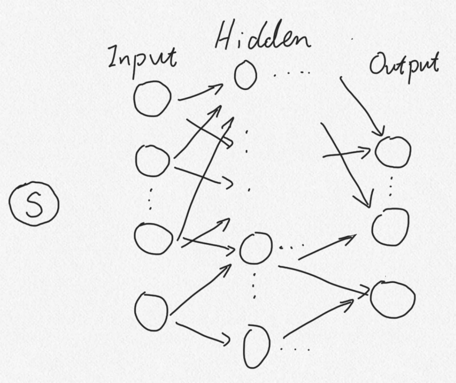




Insights from Neural net

Naive bayes: $\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}$





Can we use all data for different classes in Naive Bayes?

Correlation factor

Modify loss function: $\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j} \rightarrow$

Correlation factor

Modify loss function: $\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j} \rightarrow$

$$\log L_1(C_i, \theta) = \sum_{d \in S} \left[(y_i(d) + t) \sum_{j=1}^{v} x_j \log \theta_{i_j} \right].$$

Correlation factor

Modify loss function: $\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j} \rightarrow$

$$\log L_1(C_i, \theta) = \sum_{d \in S} \left[(y_i(d) + t) \sum_{j=1}^{v} x_j \log \theta_{i_j} \right].$$

$$\begin{aligned} \max \quad & \log L_1(C_i, \theta) \\ \text{subject to :} \quad & \sum_{j=1}^{v} \theta_{i_j} = 1; \quad \theta_{i_j} \geq 0 \end{aligned}$$

Correlation factor

Modify loss function: $\log L(C_i, \theta) = \sum_{d \in C_i} \sum_{j=1}^{v} x_j \log \theta_{i_j} \rightarrow$

$$\log L_1(C_i, \theta) = \sum_{d \in S} \left[(y_i(d) + t) \sum_{j=1}^{v} x_j \log \theta_{i_j} \right].$$

$$\begin{aligned} \max \quad & \log L_1(C_i, \theta) \\ \text{subject to :} \quad & \sum_{j=1}^{v} \theta_{i_j} = 1; \quad \theta_{i_j} \geq 0 \end{aligned}$$

$$\hat{\theta}_{i_j}^{L_1} = \frac{\sum_{d \in S} (y_i(d) + t) x_j}{\sum_{j=1}^{v} \sum_{d \in S} (y_i(d) + t) x_j}$$

v.s. original naive bayes $\hat{\theta}_{i_j} = \frac{\sum_{d \in C_i} x_j}{\sum_{d \in C_i} \sum_{l=1}^{v} x_l}$

Error analysis

Theorem

Assume $|C_i|/|S|(1 \leq i \leq k)$ *are of same order, and* $k << v$*. we have normalized length for each document, that is:* $\sum_{j=1}^{v} x_j = m$*. Then*

1. $\hat{\theta}_{i_j}^{L_1}$ *is biased, with:* $|E[\hat{\theta}_{i_j}^{L_1}] - \theta_{i_j}| = O(t)$
2. $E[|\hat{\theta}_{i_j}^{L_1} - E[\hat{\theta}_{i_j}^{L_1}]|^2] = O(\frac{1}{m|S|})$ *when* $t \approx \frac{1}{k}$.

Error analysis

Theorem

Assume $|C_i|/|S|(1 \le i \le k)$ *are of same order, and* $k << v$*. we have normalized length for each document, that is:* $\sum_{j=1}^{v} x_j = m$*. Then*

1. $\hat{\theta}_{i_j}^{L_1}$ *is biased, with:* $|E[\hat{\theta}_{i_j}^{L_1}] - \theta_{i_j}| = O(t)$
2. $E[|\hat{\theta}_{i_j}^{L_1} - E[\hat{\theta}_{i_j}^{L_1}]|^2] = O(\frac{1}{m|S|})$ *when* $t \approx \frac{1}{k}$.

Note: The order could have a large constant and lead to larger error than the original Naive bayes MSE, $\frac{\theta_{i_j}(1-\theta_{i_j})}{m|C_i|}$

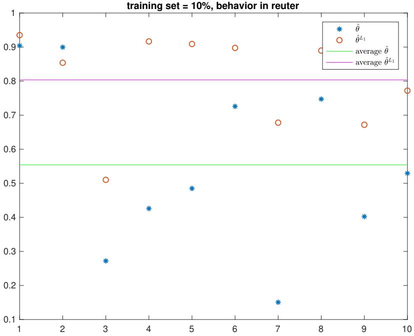
Table of Contents

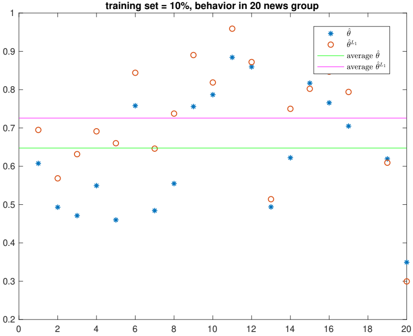
Simulation: small training set

We take two different datasets: 10 largest groups in Reuter-21578 dataset and 20 news group dataset.

Simulation: small training set

We take two different datasets: 10 largest groups in Reuter-21578 dataset and 20 news group dataset.
Small training: take 10% of the data as training set.

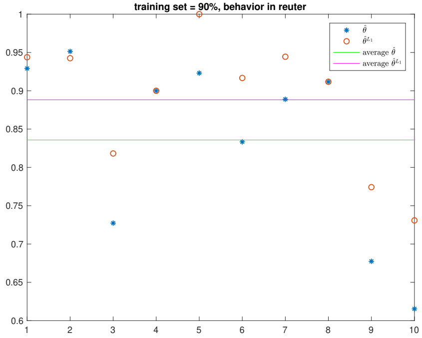




The y-axis is the accuracy, and the x-axis is the class index.

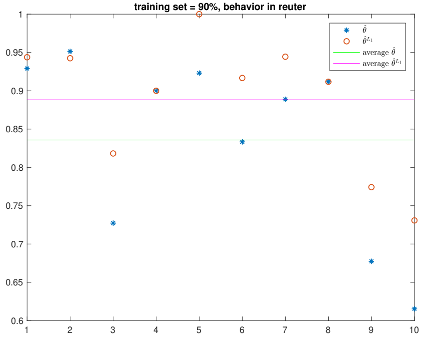
Simulation: large training set

Now take 90% of the data as training set:



Simulation: large training set

Now take 90% of the data as training set:



The bias term is dominant in 20 newsgroup.

Conclusion and future work

- Incorporate information from different classes do improve classification.

Conclusion and future work

- Incorporate information from different classes do improve classification.
- Correlation factor for Naive Bayes is better when training set is not not large.

Conclusion and future work

- Incorporate information from different classes do improve classification.
- Correlation factor for Naive Bayes is better when training set is not not large.
- Can we modify t that it adapts to different classes?

References

C. Cortes and V. Vapnik, “Support-vector networks,” *Machine learning*, vol. 20, no. 3, pp. 273–297, 1995.

T. Joachims, “Text categorization with support vector machines: Learning with many relevant features,” in *European conference on machine learning*. Springer, 1998, pp. 137–142.

N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian network classifiers,” *Machine learning*, vol. 29, no. 2-3, pp. 131–163, 1997.

P. Langley, W. Iba, K. Thompson *et al.*, “An analysis of bayesian classifiers,” in *Aaai*, vol. 90, 1992, pp. 223–228.

J. Chen, H. Matzinger, H. Zhai, and M. Zhou, “Centroid estimation based on symmetric kl divergence for multinomial text classification problem,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE, 2018, pp. 1174–1177.

D. Tang, B. Qin, and T. Liu, “Document modeling with gated recurrent neural network for sentiment classification,” in *Proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1422–1432.

P. Liu, X. Qiu, and X. Huang, “Recurrent neural network for text classification with multi-task learning,” *arXiv preprint arXiv:1605.05101*, 2016.

R. Albright, “Taming text with the svd,” *SAS Institute Inc*, 2004.

T. Hofmann, “Probabilistic latent semantic analysis,” in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*. Morgan Kaufmann Publishers Inc., 1999, pp. 289–296.

D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of machine Learning research*, vol. 3, no. Jan, pp. 993–1022, 2003.

D. D. Lewis, “Reuters-21578.”

K. Lang, “20 newsgroups data set.” [Online]. Available: `http://www.ai.mit.edu/people/jrennie/20Newsgroups/`

Thank you!